

FIND

By Chris Stramboulidis

See updates by
Larry Taylor in
LE'V2 #

Find is a machine language routine which searches your Basic program for lines which contain a specified string up to 16 characters in length. The routine is quite short (only 117 bytes) and will work with any size VZ because it resides in an unused section of the communications region.

There are two methods of entering

Find into your machine: if you have an Editor Assembler, simply type in Listing 1, set the origin to 7A28H/31272, assemble and dump the object code to tape under the name 'FIND.OBJ'. When you CLOAD or CRUN the tape, the routine will auto-run and immediately return you to the 'READY' prompt.

The other method is to type in Listing 2, which will POKE the machine code instructions into place for you and will do all the initialisation. In this case, make sure that you CSAVE a copy of 'FIND.BAS' before you try to RUN it. To save you typing it all in again if it crashes for any reason, such as a wrong number in the data

statements. A checksum is used to make sure that all these numbers add up, but this doesn't prevent numbers being placed in the wrong order. When you RUN the loader, it should only take a couple of seconds to do its job and then return you to 'READY'. The Basic loader will have been NEWed and you're ready to go.

To use Find, simply enter the following as a direct command:
PRINT&"string"

or

?&"string"

with the string to search for in between the quotes. The line numbers

of the lines which contain the search string will then be printed on the screen for you. Note that leading spaces in the search string are ignored and so the routine cannot search for spaces, eg PRINT&" " would be interpreted as a null string and would not be searched for.

PCG Apr 85 62-64

1 of 3.


```

1  ;*****
2  ;*      FIND UTILITY      *
3  ;*   FOR THE VZ-200 MICRO  *
4  ;*                               *
5  ;*      ORG=7A28H/31272    *
6  ;* SYNTAX: PRINT&"STRING" *
7  ;*                               *
8  ;*(C) 1985 C.STAMBOULIDIS*
9  ;*****
10 ;
11 BUFR EQU 7A9DH           ;BUFFER FOR SEARCH STRING
12 LEN EQU 7AD6H           ;CONTAINS LENGTH OF SEARCH STRING
13 NUM EQU 79ADH           ;CONTAINS CURRENT LINE NO.
14 NEXT EQU 79B0H          ;PT TO START OF NEXT LINE IN PST
15 ;
16 INIT LD A,0C3H          ;SET UP '&' VECTOR TO POINT
17      LD (7994H),A        ;TO OUR ROUTINE
18      LD HL,FIND
19      LD (7995H),HL
20      CALL 1B4DH          ;DO A 'NEW'
21 EXIT JP 1A19H           ;AND JUMP TO 'READY'
22 FIND INC HL             ;HL POINTS TO SEARCH STRING
23      CALL 358CH          ;MOVE STRING TO OUR BUFFER
24      LD A,(LEN)          ;GET LENGTH OF STRING
25      DEC A              ;SUBTRACT 1
26      LD (LEN),A          ;AND REPLACE IT
27      OR A               ;IF NULL STRING
28      JR Z,EXIT           ;THEN EXIT
29      LD IX,(78A4H)        ;IX=START OF PST/PTR TO NEXT LINE
30 TEST LD A,(IX+0)         ;GET LSB OF PTR
31      OR A               ;CHECK FOR ZERO
32      JR NZ,CONT          ;IF NOT, THEN CONTINUE
33      LD A,(IX+1)         ;GET MSB OF PTR
34      OR A               ;CHECK IF ZERO TOO
35      JR Z,EXIT           ;MUST BE END OF PST, SO EXIT
36 CONT LD L,(IX+0)
37      LD H,(IX+1)
38      LD (NEXT),HL        ;SAVE PTR TO NEXT LINE
39      LD L,(IX+2)
40      LD H,(IX+3)
41      LD (NUM),HL         ;SAVE CURRENT LINE NO.
42      PUSH IX             ;GET POSITION PTR
43      POP HL              ;INTO HL
44      INC HL              ;BUMP TO 1ST BYTE OF STATEMENT
45      INC HL
46      INC HL
47      INC HL
48      CALL 2B7EH          ;DE-TOKENISE CURRENT LINE

```



```

49      LD      DE,79E8H      ;DE= LOCATION OF EXPANDED LINE
50  PRE  LD      A,(LEN)      ;GET LENGTH OF SEARCH STRING
51      LD      B,A          ;INTO B
52      LD      HL,BUFR-1     ;HL= BYTE BEFORE STRING BUFFER
53  SCAN INC  HL              ;PT TO NEXT BYTE IN STRING
54      LD      A,(HL)        ;CHECK IF END OF STRING
55      OR      A
56      JR      Z,EXIT        ;IF SO, THEN WE'RE DONE
57      LD      A,(DE)        ;DE= BYTE FROM STATEMENT LINE
58      OR      A              ;CHECK FOR END OF LINE
59      JR      Z,MORE        ;IF SO, THEN PROCESS NEXT LINE
60      INC     DE             ;DE= NEXT BYTE IN STATEMENT
61      CP      (HL)          ;CHECK IF SAME AS STRING BYTE,
62      JR      NZ,PRE        ;IF NOT, THEN TRY NEXT BYTE
63      DJNZ    SCAN          ;CONTINUE UNTIL ALL BYTES FOUND
64      LD      A,20H         ;MUST BE ALL THERE, SO
65      CALL    33AH          ;PRINT A SPACE
66      LD      HL,(NUM)       ;AND PRINT THE
67      CALL    OFAFH         ;CURRENT LINE NO.
68  MORE LD      IX,(NEXT)    ;IX= PTR TO NEXT LINE IN PST
69      JR      TEST          ;BACK TO CHECK NEXT LINE

```

ERRORS : 00000

BYTES FREE :- 10288

LISTING 2

```

100  ?*****
110  ?*                               FIND.BAS                               *
120  ?*                               FIND UTILITY FOR THE VZ-200 MICRO        *
130  ?*                               ORG=7A28H/31272  SYNTAX: PRINT&"STRING"    *
140  ?*   NB. STRING LENGTH MUST BE 16 CHARACTERS OR LESS                    *
150  ?*                               (C) 1985  CHRIS STAMBOULIDIS             *
160  ?*****
170  ?
180  POKE30862,40:POKE30863,122      'SET UP USR JUMP TO INITIALISE
190  FORI=31272TO31388:READJ:C=C+J:POKEI,J:NEXT      'SET UP ROUTINE
200  IF C<>13013PRINT"CHECKSUM ERROR":STOP      'ERROR IN DATA LINES
210  X=USR(0)                          'GO INITIALISE ROUTINE
220  END
230  DATA 62,195,50,148,121,33,57,122,34,149,121,205,77,27
240  DATA 195,25,26,35,205,140,53,58,214,122,61,50,214,122
250  DATA 183,40,239,221,42,164,120,221,126,0,183,32,6,221
260  DATA 126,1,183,40,223,221,110,0,221,102,1,34,176,121
270  DATA 221,110,2,221,102,3,34,173,121,221,229,225,35,35
280  DATA 35,35,205,126,43,17,232,121,58,214,122,71,33,156
290  DATA 122,35,126,183,40,180,26,183,40,17,19,190,32,236
300  DATA 16,241,62,32,205,58,3,42,173,121,205,175,15,221
310  DATA 42,176,121,24,174

```